SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Indentify Cricket Shots

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Swayam Nikam	Exam No: B190244392
Piyush Sandhan	Exam No: B190244451
Vedant Sadgir	Exam No: B190244442
Avishkar Jadhav	Exam No: B190244308

Under The Guidance of

Prof. Bhakti Puranik

Department of Computer Engineering
DR.D.Y.PATIL INSTITUTE OF TECHNOLOGY
PIMPRI, PUNE

A.Y 2023-2024

DR.D.Y.PATIL INSTITUTE OF TECHNOLOGY
Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Entitled

Identify Cricket Shots

Submitted by

Swayam Nikam	Exam No: B190244392
Piyush Sandhan	Exam No: B190244451
Vedant Sadgir	Exam No: B190244442
Avishkar Jadhav	Exam No: B190244308

is a bonafide work carried out by Students under the supervision of Prof.Bhakti Puranik and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Prof. Bhakti Puranik
Internal Guide
Dept. of Computer Engg.

Dr. Vinod Kimbahune
H.O.D
Dept. of Computer Engg.

Dr. Lalit Wadhwa
Principal
DIT,Pimpri

Signature of Project co-ordinator

Signature of External Examiner

PROJECT APPROVAL SHEET

A Project Title

(Identify Cricket Shots)

Is successfully completed by

Swayam Nikam	Exam No: B190244392
Piyush Sandhan	Exam No: B190244451
Vedant Sadgir	Exam No: B190244442
Avishkar Jadhav	Exam No: B190244308

at

DEPARTMENT OF COMPUTER ENGINEERING

(Dr.D.Y.Patil Institute of Technology)

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2023-2024

Prof. Bhakti Puranik	Dr. Vinod Kimbahune
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

Dr.D.Y.Patil Institute of Technology, Department of Computer Engineering 2023-24 0

Abstract

Cricket shot detection is a pivotal technology transforming the landscape of the sport by providing comprehensive insights into player performance and match analysis. This abstract delves into the key components and significance of cricket shot detection systems.

The system leverages computer vision and machine learning techniques to analyze video footage of cricket matches, identifying and categorizing each shot played by batsmen with high accuracy. It extracts valuable information, including shot types (e.g., cover drive, pull, or leg glance), shot trajectory, and success rates.

Cricket shot detection finds multifaceted applications across various stakeholders in the cricketing ecosystem. For professional players and coaches, it offers an unparalleled tool for post-match analysis, aiding in performance improvement and strategic planning. Team analysts use the system to gain insights into opponents' shot patterns and formulate winning strategies.

In the broadcasting domain, shot detection enriches the viewing experience, providing real-time shot labels and engaging visualizations. Cricket enthusiasts benefit from detailed shot data, while talent scouts and cricket organizations employ the technology for player recruitment and talent development.

This abstract underscores the potential and versatility of cricket shot detection, demonstrating its transformative role in the cricket world. As the technology advances, it continues to enhance the game, empower players, and captivate audiences with a deeper understanding of the sport.

Acknowledgments

Please Write here Acknowledgment.Example given as
It gives us great pleasure in presenting the preliminary project report on ***'Indentify Cricket Shots'****.*

I would like to take this opportunity to thank my internal guide ***Prof. Bhakti Puranik*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Dr. Vinod Kimbahune****, Head of Computer Engineering Department, CollegeName for his indispensable support, suggestions.*

In the end our special thanks to ***Prof. Sunita Patil*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.*

Swayam Nikam
Piyush Sandhan
Vedant Sadgir
Avishkar Jadhav
(B.E. Computer Engg.)

Contents

List of Figures

List of Tables

Chapter 1

Synopsis

1.1 Project Title

cricket shot detection

1.2 Project Option

Detecting cricket shots in videos or live broadcasts can be a challenging but interesting project. The goal is to identify different types of cricket shots, such as drives, cuts, pulls, and more, by analyzing the video footage

1.3 Internal Guide

Prof. Bhakti Puranik

1.4 Sponsorship and External Guide

Please write if any sponsorship

1.5 Technical Keywords (As per ACM Keywords)

Cricket Analysis - The overarching theme of your project.

Computer Vision - Key technology for analyzing video frames.

Machine Learning - For training models to classify shots.

LR- A specific deep learning approach for feature extraction.

Video Processing - Dealing with video frames and real-time analysis.

Image Classification - For classifying frames into different shot types.

Real-time Processing - The ability to analyze shots as they happen.

Feature Extraction - Identifying important characteristics in video frames.

Shot Recognition - Identifying and classifying cricket shots.

Sports Analytics - Using data analysis techniques for sports performance evaluation.

Action Recognition - Detecting specific actions (shots) within a video.

Computer Graphics - If you're overlaying labels or enhancing the visual presentation.

Pattern Recognition - For recognizing patterns in cricket shots.

Deep Learning - The broader category of machine learning techniques, including CNNs.

Motion Analysis - Analyzing the movement of the ball and players.

Real-time Applications - Focusing on the live aspect of your system.

Data Annotation - Preparing the dataset with labeled shot types.

Model Evaluation - Assessing the performance of your classification model.

Human-Computer Interaction - If you're developing a user interface.

Video Surveillance - Relevant to the real-time aspect of shot detection.

1.6 Problem Statement

"In recent years, several sports have gained more popularity and attention. Numerous people were hankering to watch some sporting event because there was no sporting event during the current pandemic. Cricket is without a doubt one of the most popular sports in India, with millions of devoted fans".

1.7 Abstract

- Numerous sports have attracted a lot of interest and popularity recently. The absence of any sporting activities throughout the height of the current outbreak had caused a sizable number of people to crave

to see some game being played. In India, cricket is undoubtedly the most well-liked sport, with millions of fans who passionately follow the matches. As a result of their intense interest in the game, fans analyse each player's skills in great detail, especially their shot selection. The popularity of fantasy leagues and other services of a similar nature has increased interest in assessing players so that they might be chosen for their teams. The manual process for identifying batter shots is one of the most time-consuming. and time-consuming procedures

1.8 Goals and Objectives

- The suggested method for Cricket Shot recognition and analysis is created on a Windows-based workstation using the Python computer language.
- This tactic is programmed using the Spyder IDE. The deployment computer has a 1TB hard drive, 8GB of RAM, and an Intel Core i5 processor.
- The Cricket shot recognition and analysis technique's dependability needs to be assessed for the Convolution neural network deployment to be successful. This technique makes use of a photograph of five different cricket shots made by left- and right-handed batsmen..

1.9 Relevant mathematics associated with the Project

System Description:

1.10 Mathematical Model

- Mathematical Model
- **Let S be the Whole system S : (I,P,O)**
  - I-input
  - P-procedure
  - O-output

- **Input(I)**
- I = (Input as Live camera)
- Where,
- Live camera : for capturing players activity .
- **Procedure (P)**
- P = (I, Using I System perform operations and detect shots of cricket)
- **Output(O)**
- O = (System detect the players cricket shots.)

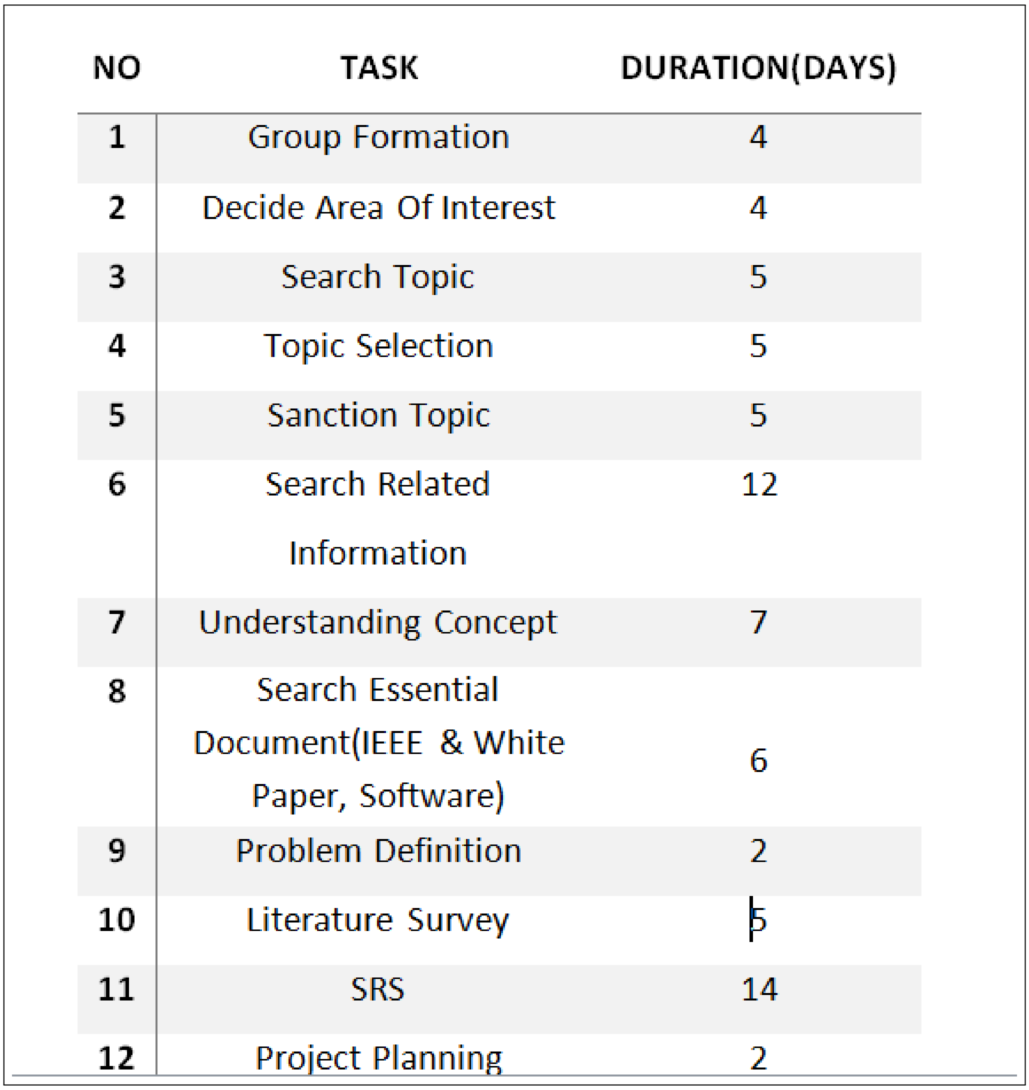
1.11 Names of Conferences / Journals where papers can be published

- IEEE/ACM Conference/Journal 1
- Conferences/workshops in IITs
- Central Universities or SPPU Conferences
- IEEE/ACM Conference/Journal 2

1.12 Review of Conference/Journal Papers supporting Project idea

Atleast 10 papers + White papers or web references
Brief literature survey [Description containing important description of at least 10 papers

1.13 Plan of Project Execution



NO	TASK	DURATION(DAYS)
1	Group Formation	4
2	Decide Area Of Interest	4
3	Search Topic	5
4	Topic Selection	5
5	Sanction Topic	5
6	Search Related Information	12
7	Understanding Concept	7
8	Search Essential Document(IEEE & White Paper, Software)	6
9	Problem Definition	2
10	Literature Survey	5
11	SRS	14
12	Project Planning	2

Chapter 2

Technical Keywords

2.1 Area of Project

The project area for ”Cricket Shot Detection” falls within the intersection of several key fields in computer science and technology

2.2 Technical Keywords

1. Cricket Analysis - The overarching theme of your project.
2. Computer Vision - Key technology for analyzing video frames.
3. Machine Learning - For training models to classify shots.
4. LR- A specific deep learning approach for feature extraction.
5. Video Processing - Dealing with video frames and real-time analysis.
6. Image Classification - For classifying frames into different shot types.
7. Real-time Processing - The ability to analyze shots as they happen.
8. Feature Extraction - Identifying important characteristics in video frames.
9. Shot Recognition - Identifying and classifying cricket shots.
10. Sports Analytics - Using data analysis techniques for sports performance evaluation.
11. Action Recognition - Detecting specific actions (shots) within a video.

12. Computer Graphics - If you're overlaying labels or enhancing the visual presentation.

13. Pattern Recognition - For recognizing patterns in cricket shots.

14. Deep Learning - The broader category of machine learning techniques, including CNNs.

15. Motion Analysis - Analyzing the movement of the ball and players.

16. Real-time Applications - Focusing on the live aspect of your system.

17. Data Annotation - Preparing the dataset with labeled shot types.

18. Model Evaluation - Assessing the performance of your classification model.

19. Human-Computer Interaction - If you're developing a user interface.

20. Video Surveillance - Relevant to the real-time aspect of shot detection.

Chapter 3

Introduction

3.1 Project Idea

- The project area for a cricket shot detection system can be broadly categorized as "Computer Vision and Sports Analytics."

3.2 Motivation of the Project

- Recent years have seen significant advancements in cricket batting, with a focus on the game's shorter formats1. With the development of advanced analysis and machine learning, there has been an increased demand to maximise performance and achievement at the greatest level. With an emphasis on analysis, the rising use of technology in conjunction with science and medicine has been dubbed a game-changer in the athletic world. Sports video footage analysis has experienced considerable advancements thanks to the development of artificial intelligence, deep learning, and multimedia technologies.

3.3 Literature Survey

1.Paper Name: CRICKET SHOT DETECTION FROM VIDEOS
Author: Archit Semwal, Durgesh Mishra, Vineet Raj, Jayanta Sharma and Ankush Mittal
Abstract : Classifying various type of bat strokes played in a cricket match has always been an arduous undertaking while indexing the cricket sport. Identifying the type of shot played by the batsman in a cricket match is a substantial aspect as well as one of the unplumbed subjects in this domain. This paper proposes a novel scheme to recognize and classify different types of bat shots played in cricket. The model relies on the state-ofthe-art techniques like saliency and optical flow to bring out static and dynamic cues and on Deep Convolutional Neural Networks (DCNN) for extracting representations. Moreover, a completely new dataset of 429 videos, has been introduced to evaluate the performance of the proposed framework. The model achieves an accuracy of 83.098percent for three classes of right-handed shots and 65.186percent for three classes of left-handed shots.

- **2.Paper Name:** :- Automatic Video Summarization from Cricket Videos Using Deep Learning
  Author: Solayman Hossain Emon, A.H.M Annur, Abir Hossain Xian, Kazi Mahia Sultana4 and Shoeb Mohammad Shahriar
  Abstract : Video Summarization is a technique that reduces the viewing duration of a video by extracting important parts from a large video. As most of the cricket matches are long-lasting, the audience preferably needs to have the summarized format of the large content. In this paper, we intend to speed up large-scale video processing by providing a concise synopsis that still conveys the story of a cricket match. For addressing this challenge, we develop a Deep Cricket Summarization Network (DCSN) that automatically extracts key-shots from an input video. Due to the limitations of the few available datasets in this domain, we introduce a new dataset named CricSum. The quality of the generated summary highly relies on users' subjective understandings. So we demonstrate the efficiency of the proposed summarization system using the Mean Opinion Score (MOS) technique. The automatically generated summarized videos collectively obtained MOS value 4 out of 5.

- **3.Paper Name :** Deep CNN based Data-driven Recognition of Cricket Batting Shots
  Author name: Muhammad Zeeshan Khan, Muhammad A. Hassan, Ammarah Farooq
  abstract : Cricket is one of the most played and watched sports, specially in the South Asian region. This paper deals with identifying and categorizing various batting shots from cricket videos. Proposed method is based on deep convolution neural networks. Results have been evaluated for both 2D convolution followed by recurrent network for processing sequence of video frames and 3D convolution network for capturing spatial and temporal features simultaneously. In order to train and evaluate models, dataset comprising of about 800 batting shot clips have been locally developed. Obtained models are able to recognize a shot being played with 90percent accuracy. The distinction of such visually similar shots with this high accuracy is novel in literature and indicates the high implications of modern AI and deep learning in applications for detecting various cricket activities as well as for decision making purposes. The prepared dataset will be made publicly available for research community.

- **4.Paper Name :** Extraction of Strong and Weak Regions of Cricket Batsman through Text-Commentary Analysis
  Author::- Muhammad Arslan Rauf, Haseeb Ahmad, CM Nadeem Faisal
  abstract : Cricket is a famous game in the world where many metrics are introduced and being used to help the coaches and umpires to solve the critical problems. Though different statistics are used to quantify the player's performance based on strike rate, average or for ranking, prediction, and optimal team selection. There is not any effective method to measure the strong and weak regions of a batsman in cricket. In this paper, a text mining method is presented to extract either the strong shot selection points that are frequent for scoring or weak shot regions that seem tough for a batsman to play. Also, a mechanism is put forward to calculate the region-wise strike rate of an individual batsman. To achieve the objectives, the T20 cricket text commentary is being used for this purpose which is available on the espncricinfo website. The proposed method can be helpful for coaches and players to know the strong or weak regions where the batsman feels ease or difficulty to play, respectively. Moreover, the opponent bowlers can also use this method to plan the area where to bowl to each batsman.

- **5.Paper Name :** AN INTERACTIVE APPROACH TO IDENTIFY CRICKET SHOTS THROUGH DEEP LEARNING MECHANISM
  Author: Atharv Pushkraj Nirgude, Rohit Dinesh Sonone, Sahil Vikram Sonawane, Rushikesh Sham Ahire, Prof. Balaji Bodkhe
  Abstract: In recent years, numerous sports have received significant attention and appeal. In the height of the current epidemic, the unavailability of any athletic events had left a significant number of people yearning to observe some game being played. Cricket is arguably the most popular sport in India, with millions of followers who watch the games obsessively. Fans are enthralled by the game and conduct in-depth analyses of the individual players and their abilities, particularly their shot choices. With the growth of fantasy leagues as well as other similar services, there is a growing interest in evaluating individuals who are playing well so that they may be selected for their teams. The manual procedure of batter shot recognition is amongst the most time-consuming and labor-intensive processes that might be automated. As a result, this study provides a successful cricket shot assessment approach that makes use of deep learning in the form of Convolutional Neural Networks to fulfill its objectives.

- **6.Paper Name :** Improving Deep Learning Performance Using Random Forest HTM Cortical Learning Algorithm
  Author: Mohamed AbdElhamid Abbas , PhD
  Abstract : Deep Learning is a feature of artificial intelligence that mimics how the human brain processes data and creates forms for use in selection construction. The paper's goal is to enhance deep learning performance utilising the RFHTMC algorithm, which has been proposed. This suggested algorithm combines the Random Forest and HTM Cortical Learning Algorithm versions. The idea of minimising the mean absolute percentage error, which is a sign of the high performance of the prediction procedure, is the foundation of the methodology for increasing Deep Learning performance. Along with the overlap duty cycle, the classifier's processing speed is also indicated by the classifier's high percentage of processing time. The results show that the suggested set of rules

- **7.Paper Name:** Random Forest HTM Cortical Learning Algorithm
  Author: AbdElhamid Abb
  Abstract : Deep Learning is an artificial intelligence function that imitates the mechanisms of the human mind in processing records and developing shapes to be used in selection construction.The objective of the paper is to improve the performance of the deep learning using a proposed algorithm called RFHTMC. This proposed algorithm is a merged version from Random Forest and HTM Cortical Learning Algorithm. The methodology for improving the performance of Deep Learning depends on the concept of minimizing the mean absolute percentage error which is an indication of the high performance of the forecastprocedure. In addition to the overlap duty cycle which its high percentage is an indication of the speed of the processing operation of the classifier.

- **8.Paper Name :** DeepRT: A Predictable Deep Learning Inference Framework for IoT Devices
  Author name: Woochul Kang and Daeyeon Kim
  abstract : Deep learning is already transforming how computers integrated into IoT devices use sensor feeds to make wise decisions in the real world. For mobile and IoT devices with limited resources, there have been major efforts to design lightweight and extremely efficient deep learning inference techniques. Some approaches suggest a hardware-based accelerator, while others suggest employing various model compression techniques to reduce the amount of computing required for deep learning models [1]. Despite the fact that these initiatives have shown appreciable improvements in performance and efficiency, they are unaware of the Quality-of-Service (QoS) requirements of different IoT applications and as a result, exhibit unpredictable "best-effort" performance in terms of inference latency, power consumption, resource usage, etc. Unpredictability in IoT devices with temporal constraints may have negative consequences, including

- **9.Paper Name :** cognitive foundation of knowledge science and deep knowledge learning by cognitive robots
  Author::- yingxu wang
  abstract : The comprehension of natural intelligence and the development of appropriate mathematical tools for rigorously modelling the brain in forms that are machine comprehensible are deeply rooted in one another, according to recent basic studies. Learning is the cognitive process of acquiring knowledge and action. Object identification, cluster categorization, functional regression, behaviour production, and knowledge acquisition are the five categories that learning can be divided into. According to Wang's most recent work in knowledge science, the fundamental unit of knowledge is a binary relation (bir), such as a bit for information and data. The field of cognitive machine learning has emerged in response to a basic problem with knowledge learning that is distinct from those presented by deep and recurrent neural network technologies.

- **10.Paper Name :** Structure of Deep Learning Inference Engines for Embedded Systems
  Author: Seung-mok Yoo, Changsik Cho, Kyung Hee Lee
  Abstract: Numerous deep learning applications have been released during the past few years. On servers or PCs with GPUs, the majority of deep learning-related research and development has been conducted. Several initiatives to apply those applications to the industrial sector have recently been made. Contrary to the laboratory setting, there may be some geographical and environmental limitations when applying deep learning techniques to real-world objectives. In this paper, we outline the requirements for running deep learning programmes on embedded platforms. Here, we describe our ongoing work on creating a deep learning framework for embedded systems, particularly those found in automobiles. Typically, the process of developing a deep learning application may be broken down into two steps: training a data model with a large data set and using the data model with

Chapter 4

Problem Definition and scope

4.1 Problem Statement

"In recent years, several sports have gained more popularity and attention. Numerous people were hankering to watch some sporting event because there was no sporting event during the current pandemic. Cricket is without a doubt one of the most popular sports in India, with millions of devoted fans".

4.1.1 Goals and objectives

Goal and Objectives:

- The suggested method for Cricket Shot recognition and analysis is created on a Windows-based workstation using the Python computer language.

- This tactic is programmed using the Spyder IDE. The deployment computer has a 1TB hard drive, 8GB of RAM, and an Intel Core i5 processor.

- The Cricket shot recognition and analysis technique's dependability needs to be assessed for the Convolution neural network deployment to be successful. This technique makes use of a photograph of five different cricket shots made by left- and right-handed batsmen..

4.1.2 Statement of scope

- Develop a real-time cricket shot detection system capable of recognizing and classifying various types of cricket shots, such as drives, cuts, pulls, and sweeps.
- Create a user-friendly interface for visualizing shot detections during live broadcasts and post-game analysis.
- Evaluate the performance of the shot detection system using machine learning models and real-world cricket match data.
- Collaborate with cricket organizations to explore deployment possibilities for enhancing sports analytics and viewers' experiences.

4.2 Major Constraints

- Data Availability: Availability of high-quality and diverse cricket video data is crucial for training and testing the shot detection system. Limited or low-quality data can significantly impact the system's accuracy and generalization.
- Lighting and Environmental Factors: Cricket matches are played in various lighting conditions and environments, including daylight, floodlights, and changing weather conditions. These variations can make shot detection more challenging, as lighting changes may affect the appearance of the shots.
- Camera Angles and Perspectives: Different camera angles and perspectives in cricket broadcasts can make it difficult to consistently capture shots. Variability in camera placement and angle can affect the visibility and recognition of shots.

4.3 Outcome

- Accurate Shot Detection: The primary outcome is the ability to accurately detect and classify various cricket shots, such as drives, cuts, pulls, and sweeps, in real-time and post-game analysis. The system should provide reliable results for different players and match conditions.

- Real-Time Application: The system should be capable of real-time shot detection, making it suitable for live cricket broadcasts and immediate analysis. This enhances the viewing experience and provides valuable insights to sports analysts.
- User-Friendly Interface: A user-friendly interface for visualizing shot detections should be available, allowing broadcasters, analysts, and viewers to easily understand and interact with the data. This outcome enhances the accessibility and usability of the system.

4.4 Applications

- The process described in this research paper proposes a technique for automatically analyzing cricket shots that uses deep learning strategies to reach a significantly better result.
- Convolutional Neural Networks are used in the described technique to utilize the video input of each cricket shot being played. A dataset of numerous distinct shots performed by a batsman is used to train the CNN model.
- To perform the cricket shot assessment, the results are efficiently categorized utilizing the decision-making process.
- The outcomes achieved by the prescribed approach are outlined in the research article in the results and discussion sections

4.4.1 Hardware Interfaces:

RAM : 8 GB
As we are using Machine Learning Algorithm and Various High Level Libraries Laptop
RAM minimum required is 8 GB.
Hard Disk : 40 GB
Data Set of CT Scan images is to be used hence minimum 40 GB Hard Disk memory is required.
Processor : Intel i5 Processor
Pycharm IDE that Integrated Development Environment is to be used and data loading should be fast hence Fast Processor is required
IDE : Spyder

Best Integrated Development Environment as it gives possible suggestions at the time of typing code snippets that makes typing feasible and fast.
Coding Language : Python Version 3.8
Highly specified Programming Language for Machine Learning because of availability of High Performance Libraries.
Operating System : Windows 10
Latest Operating System that supports all type of installation and development Environment

4.4.2 Software Interfaces

Operating System: Windows 10

IDE: Spyder

Programming Language : Python

Chapter 5

Project Plan

5.1 Project Estimates

5.1.1 Reconciled Estimates

1.Requirement gathering and analysis: In this step of waterfall we identify what are various requirements are need for our project such are software and hardware required, database, and interfaces.
2.System Design: In this system design phase we design the system which is easily understood for end user i.e. user friendly. We design some UML diagrams and data flow diagram to understand the system flow and system module and sequence of execution.
3.Implementation: In implementation phase of our project we have implemented various module required of successfully getting expected outcome at the different module levels. With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
4.Testing: The different test cases are performed to test whether the project module are giving expected outcome in assumed time. All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
5.Deployment of System: Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
6.Maintenance: There are some issues which come up in the client en-

vironment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.
All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

5.1.2 Project Resource

Well configured Laptop, eclipse IDE, 2 GHZ CPU speed, 2 GB RAM, Internet connection

5.2 Risk Management

1.In appropriate dataset -To overcome this risk we are trying to use well organized and complete dataset.
2.Security- To overcome and improving security we use multilevel security like access permissions of user.

5.2.1 Risk Identification

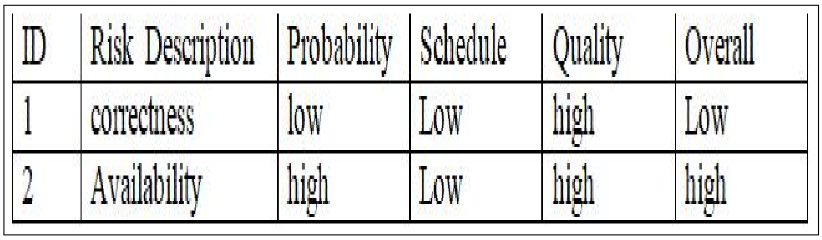
1. Have top software and customer managers formally committed to support the project?
Ans-Not applicable.
2. Are end-users enthusiastically committed to the project and the system/product to be built?
Ans-Not known at this time.
3. Are requirements fully understood by the software engineering team and its customers?
Ans-Yes
4. Have customers been involved fully in the definition of requirements?
Ans-Not applicable
5. Do end-users have realistic expectations?
Ans-Not applicable
6. Does the software engineering team have the right mix of skills?
Ans-yes
7. Are project requirements stable?
Ans-Not applicable

8. Is the number of people on the project team adequate to do the job?
Ans-Not applicable
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
Ans-Not applicable

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

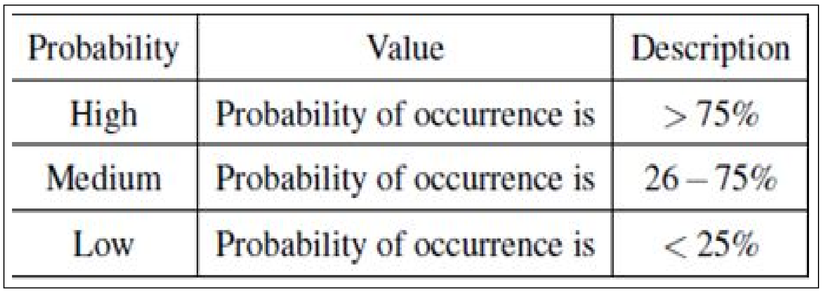
5.2.3 Risk Table



ID	Risk Description	Probability	Schedule	Quality	Overall
1	correctness	low	Low	high	Low
2	Availability	high	Low	high	high

Figure 5.1

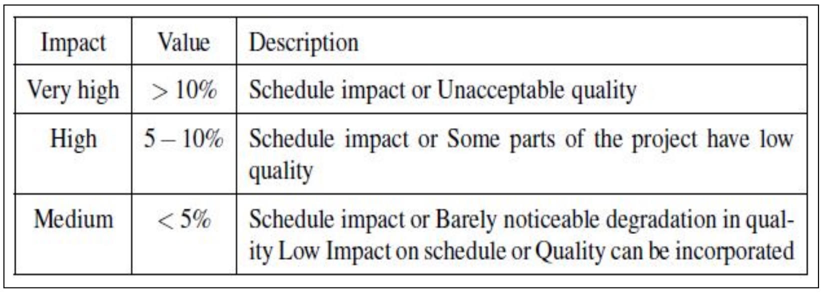
Risk Probability definitions:



Probability	Value	Description
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 − 75%
Low	Probability of occurrence is	< 25%

Figure 5.2

Risk Impact definitions:

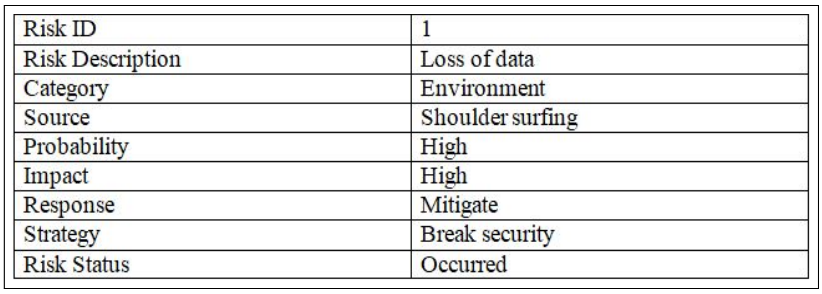


Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Figure 5.3

5.2.4 Overview of Risk Mitigation, Monitoring, Management:

Following are the details for each risk.



Risk ID	1
Risk Description	Loss of data
Category	Environment
Source	Shoulder surfing
Probability	High
Impact	High
Response	Mitigate
Strategy	Break security
Risk Status	Occurred

Figure 5.4

5.3 Project Schedule

5.3.1 Project Task Set

Major Tasks in the Project stages are:

- Task 1: correctness
- Task 2: availability
- Task 3: integrity

5.3.2 Task Network

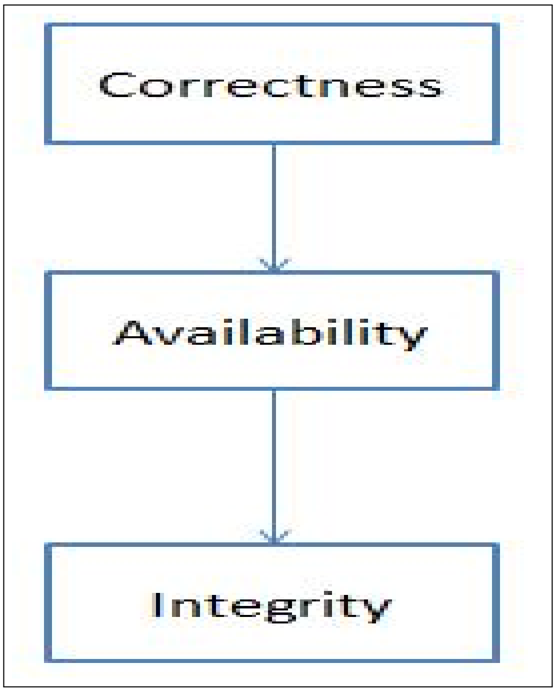


Figure 5.5: Task Network

5.3.3 Timeline Chart

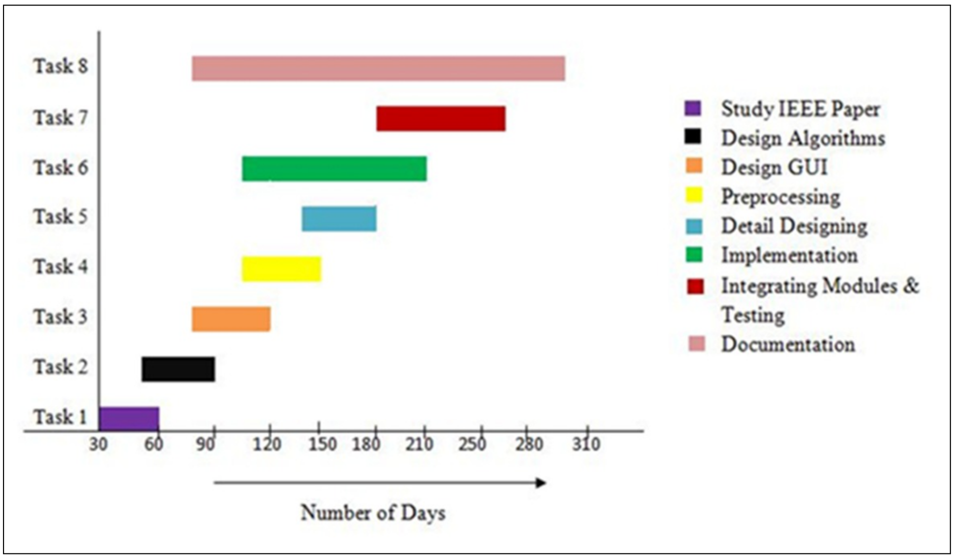


Figure 5.6: Timeline Chart

5.4 Team Organization

Team consists of 4 members and proper planning mechanism are used and roles of each member are defined.

5.4.1 Team Structure

The team structure for the project is identified. There are total 4 members in our team and roles are defined. All members are contributing in all the phases of project.

5.4.2 Management Reporting And Communication

Well planning mechanisms are used for progress reporting and inter/intra team communication are identified as per requirements of the project.

Chapter 6

Software requirement specification

6.1 Introduction

Cricket, often described as the ”gentleman’s game,” has evolved into a dynamic sport filled with breathtaking displays of skill, strategy, and athleticism. At the heart of this evolution lies the artistry of shot-making, where batsmen gracefully wield the willow to send the cricket ball soaring across the field. The ability to discern and classify these cricket shots, from the elegant cover drive to the audacious pull, holds immense value in understanding the nuances of the game.

In the age of technology and data-driven sports analysis, the development of a real-time cricket shot detection system represents a pivotal step in unlocking the secrets hidden within the pitch. This project endeavors to harness the power of computer vision and machine learning to accurately and swiftly identify and categorize various cricket shots, transforming the way we view, analyze, and interact with the game.

6.1.1 Purpose and Scope of Document

The purpose of this document is to provide a comprehensive project proposal for the development of a real-time cricket shot detection system. This document serves as a foundational blueprint for the project, offering a detailed understanding of its objectives, methodologies, and anticipated outcomes. It is intended to guide project stakeholders, including team members, cricket organizations, potential collaborators, and investors, in understanding the project’s scope, significance, and technical aspects.

6.1.2 Overview of responsibilities of Developer

Software Development: Develop and maintain the software components of the cricket shot detection system, including data preprocessing, machine learning models, and real-time processing.

Algorithm Design: Design and implement computer vision and machine learning algorithms for shot detection, feature extraction, and model training.

Data Processing: Develop routines for data collection and preprocessing, such as video frame extraction, resizing, and normalization.

Machine Learning Modeling: Build and fine-tune machine learning models, such as convolutional neural networks (CNNs), for shot classification.

Real-Time Processing: Implement real-time shot detection algorithms to analyze live cricket matches and broadcasts.

Model Optimization: Optimize machine learning models for performance and efficiency, considering factors like model size and speed.

User Interface Development: Create a user-friendly interface for visualizing shot detections and interacting with the system, which may involve frontend development.

Testing and Quality Assurance: Test the system for accuracy, robustness, and performance. Identify and address software bugs and issues.

Data Annotation: Collaborate with data annotators to ensure the dataset is accurately labeled for training machine learning models.

Research and Innovation: Stay updated with the latest developments in computer vision and machine learning to apply cutting-edge techniques to the project.

Documentation: Maintain comprehensive documentation for the project, including code comments, user manuals, and technical documentation.

Collaboration: Collaborate with other team members, data scientists, domain experts, and cricket organizations to ensure the project's success.

Project Management: Contribute to project planning, task estimation, and tracking progress to meet project deadlines.

Continual Learning: Actively engage in ongoing learning and skill development to adapt to evolving technologies and project requirements.

Communication: Communicate project progress, challenges, and solutions with team members, stakeholders, and project leads.

Security and Privacy: Ensure data security and privacy compliance in data handling and processing.

Deployment and Support: Assist in the deployment of the system for live cricket broadcasts or post-game analysis, and provide support and maintenance as needed.

Adherence to Best Practices: Follow coding standards, best practices, and quality assurance processes to maintain a high standard of code quality.

6.2 Usage Scenario

These usage scenarios illustrate the broad spectrum of applications and benefits that a cricket shot detection system can offer to players, coaches, broadcasters, analysts, fans, and various stakeholders in the world of cricket. The system's capabilities can transform how cricket is played, watched, and analyzed, fostering a data-driven approach to the sport.

6.2.1 User profiles

Understanding these user profiles helps in tailoring the system's features and capabilities to meet the diverse needs of players, coaches, broadcasters, analysts, fans, and various stakeholders involved in the world of cricket. Each user profile represents a different facet of the system's utility and its potential to transform the way cricket is played, analyzed, and enjoyed.

6.2.2 Use-cases

Use-Case 1: Player Performance Analysis

Actor: Professional Cricket Player

Description: The player uses the system to analyze their performance.

Tasks:

The player uploads videos of their recent matches.

The system extracts and processes video frames to identify and classify each shot played by the player.

The player reviews a performance report that includes shot types, success rates, and areas for improvement.

Use-Case 2: Coaching and Training

Actor: Cricket Coach

Description: The coach uses the system to evaluate players and provide targeted training.

Tasks:

The coach uploads videos of players' performances.

The system analyzes the shots and provides performance metrics.

The coach uses the insights to create tailored training programs and offer specific guidance to players. Use-Case 3: Match Strategy D

6.2.3 Use Case View

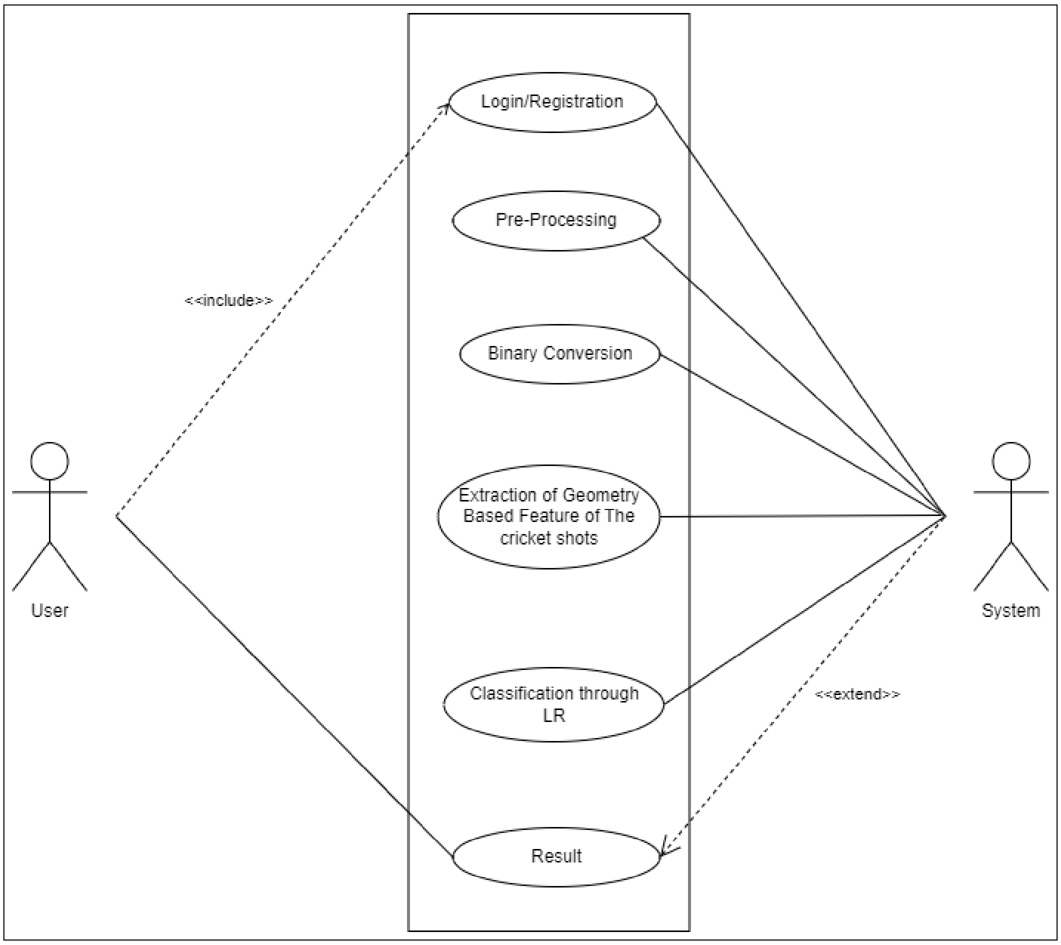


Figure 6.1: Use case diagram

6.3 Data Model and Description

6.3.1 Data Description

Describing and documenting data is essential in ensuring that the researcher, and others who may need to use the data, can make sense of the data and understand the processes that have been followed in the collection, processing, and analysis of the data. Research data are any physical and/or digital materials that are collected, observed, or created in research activity for purposes of analysis to produce original research results or creative works.

6.3.2 Data objects and Relationships

A data object is a part of the repository whose content can be addressed and interpreted by the program. All data objects must be declared in the ABAP program and are not persistent, meaning that they only exist while the program is being executed. Before you can process persistent data (such as data from a database table or from a sequential file), you must read it into data objects first. Conversely, if you want to retain the contents of a data object beyond the end of the program, you must save it in a persistent form.

6.4 Functional Model and Description

6.4.1 Data Flow Diagram

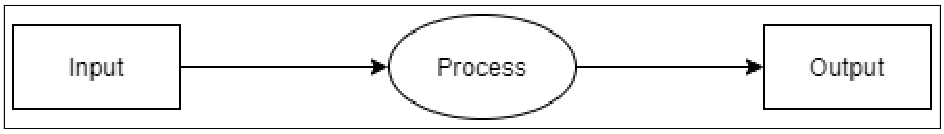


Figure 6.2: Data Flow(0) diagram

6.4.1.1 Level 1 Data Flow Diagram

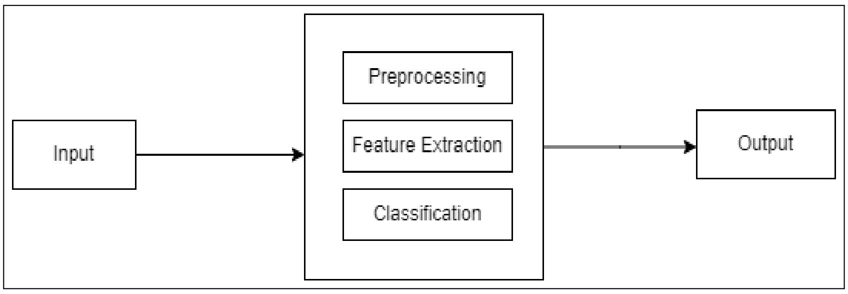


Figure 6.3: Data Flow(1) diagram

6.4.1.2 Level 2 Data Flow Diagram

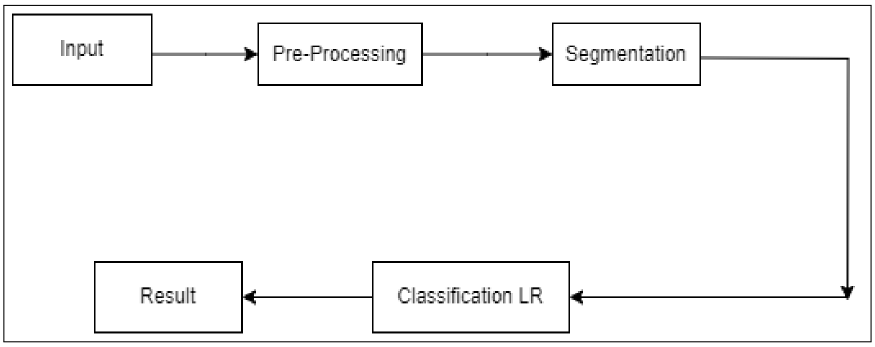


Figure 6.4: Data Flow(2) diagram

6.4.2 Activity Diagram:

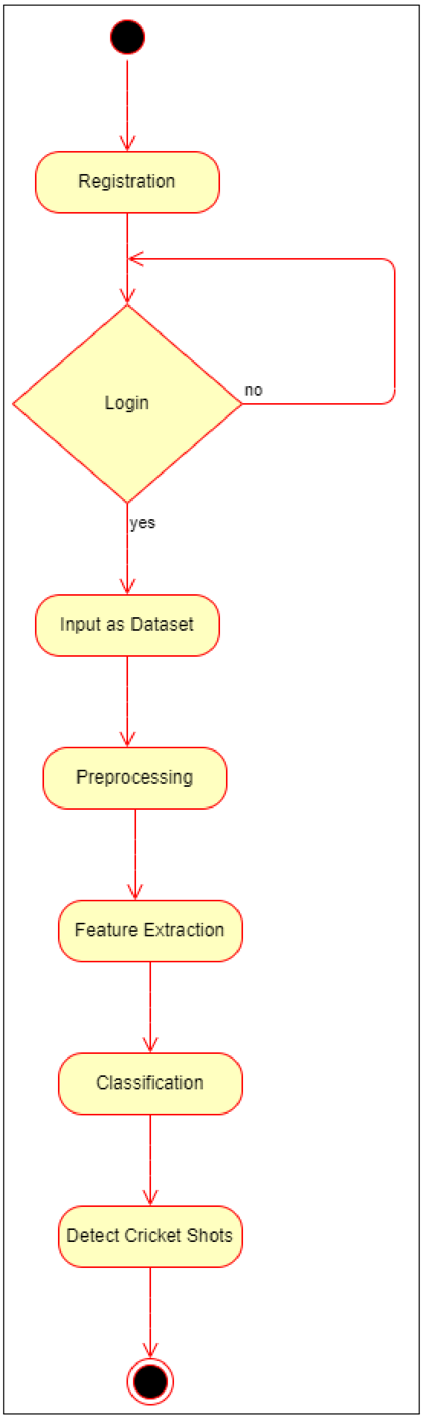


Figure 6.5: Activity Diagram

6.5 NON FUNCTIONAL REQUIREMENT

6.5.1 PerformanceRequirements

• The performance of the functions and every module must be well.

• The overallperformance of the software will enable the users to work efficiently.

• Perfor-mance of encryption of data should be fast.

• Performance of the providing virtual environment should be fast

6.5.2 Safety Requirement

• The application is designed in modules where errors can be detected and fixed easily.

•This makes it easier to install and update new functionality if required.

6.5.3 Software Quality Attributes

Our software has many quality attribute that are given below:-

- Adaptability: This software is adaptable by all users.

- Availability: This software is freely available to all users. The availability of the software is easy for everyone.

- Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

- Reliability: The performance of the software is better which will increase the reliabilityof the Software.

- User Friendliness: Since, the software is a GUI application; the output generated is much user friendly in its behavior.

- Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.

- Security: Users are authenticated using many security phases so reliable security is provided.

- Testability: The software will be tested considering all the aspects.

6.5.4 State Diagram:

State Transition Diagram
A state diagram, also known as a state machine diagram, is a graphical representation of a system's behavior over time. It shows the various states a system or object can be in and how transitions between states occur in response to events or conditions. For a cricket shot detection system, a simplified state diagram might include the following states and transitions

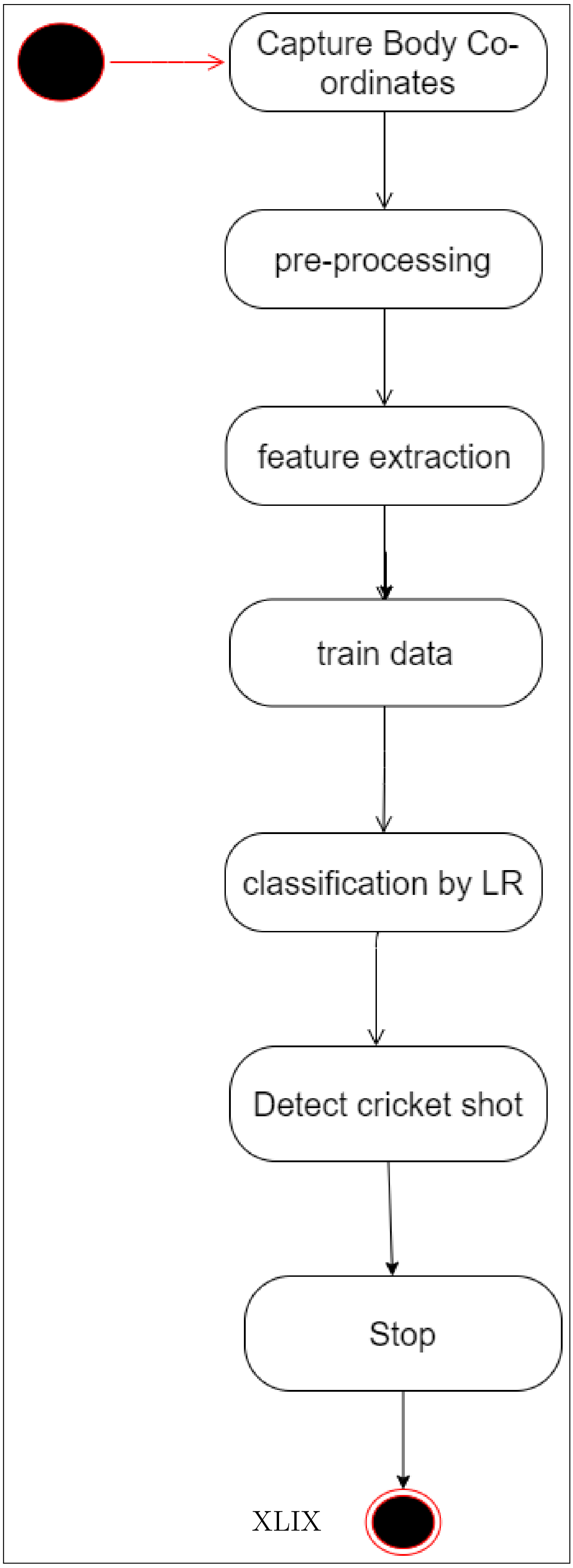


Figure 6.6: State transition diagram

6.5.5 Design Constraints

Data Availability and Quality: The system's performance is highly dependent on the quality and availability of cricket video data. Limited or poor-quality data can constrain the system's accuracy and effectiveness.

Lighting and Environmental Variability: Cricket matches are played in various lighting conditions, including daylight, floodlights, and different weather conditions. Adapting to these variations is a design challenge.

6.5.6 Software Interface Description

User Interface: The user interface provides an interactive platform for users to interact with the system. It may include web applications, mobile apps, or desktop applications. Users can request shot analysis, view results, and configure system settings through this interface.

Data Sources: Data sources include video feeds or recorded cricket match videos. The system needs to interface with these sources to obtain the raw video data for shot detection.

Data Preprocessing Module: Before shot detection can occur, the system interfaces with a data preprocessing module that extracts individual frames from the video, normalizes lighting conditions, and prepares the data for analysis.

Machine Learning Model: The core of the system interfaces with the machine learning model, which is responsible for shot detection. This involves sending video frames to the model for shot classification and receiving the results.

Chapter 7

Detailed Design Document using Appendix A and B

7.1 Architectural Design

Description of the problem related to the product.Details about the product, including its purpose, features, and any relevant technical specifications.Specific requirements or goals that the design should achieve.Any constraints or limitations that need to be considered.Stakeholders involved in the project.Any specific guidelines or standards that must be followed.

7.2 Data design (using Appendices A and B)

7.3 Data design

A description of all data structures including internal, global, and temporary data structures, database design (tables), le formats.

7.4 Internal software data structure

Stack, Queue.

7.5 Global data structure

Hashmap.

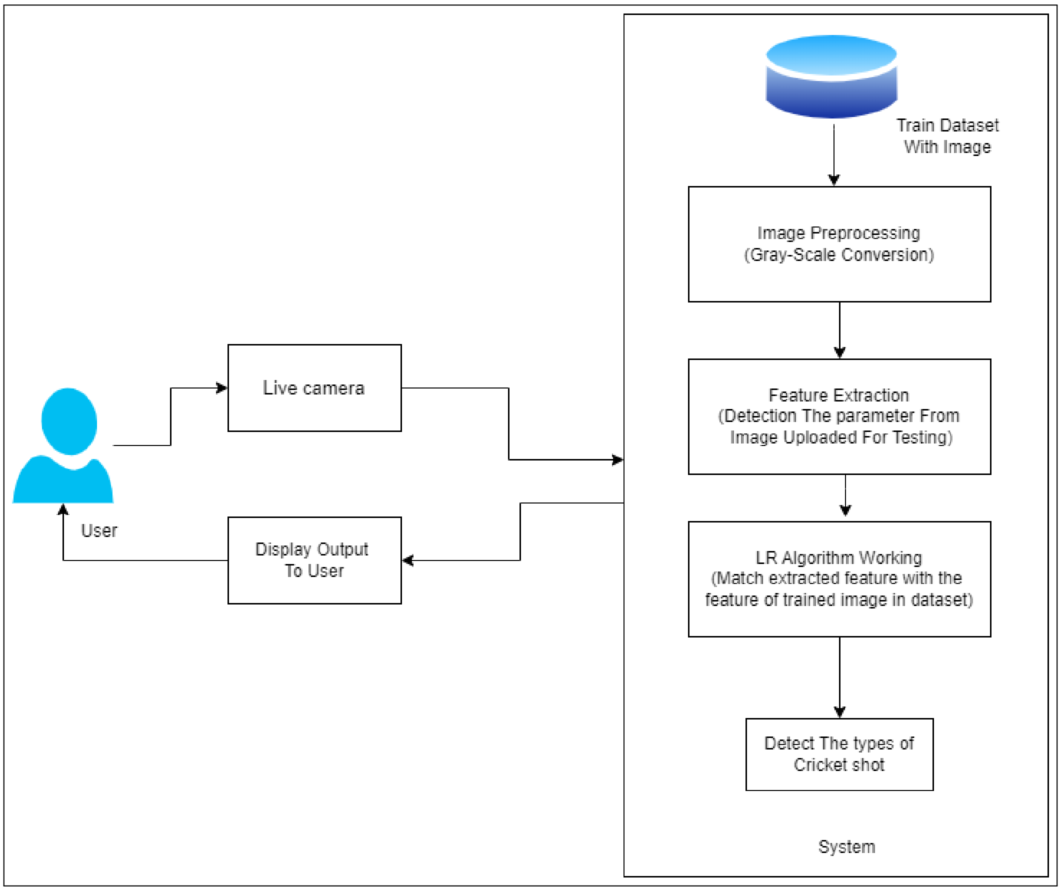


Figure 7.1: Architecture diagram

7.6 Temporary data structure

Stack, Queue.

7.7 Database description

MySQL: MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software. MySQL is a database management system. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

7.8 Compoent Design

Class diagrams, Interaction Diagrams, Algorithms. Description of each component description required.

7.8.1 Class Diagram

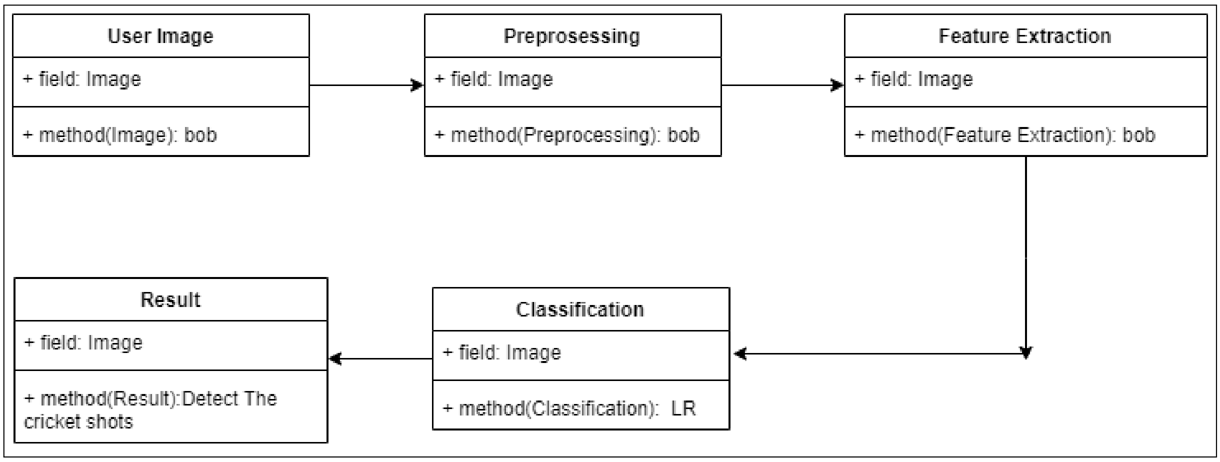


Figure 7.2: Class Diagram

Chapter 8

Project Implementation

8.1 Introduction

Project Overview: Briefly describe the project, its purpose, and its importance. Provide a concise summary of the project's background, objectives, and the problem it aims to solve.

Implementation Objectives: Outline the specific goals and objectives of the project implementation phase. What are we looking to achieve during this stage? What milestones and deliverables can we expect?

Scope of Implementation: Define the scope of the implementation phase. What aspects of the project will be addressed, and what will be the boundaries of this phase? This helps set clear expectations for the reader.

Significance of Implementation: Explain why the implementation phase is crucial for the project's success. What benefits or outcomes are expected as a result of effective implementation?

Structure of This Section: Provide an overview of what the reader can expect in this section. Mention the key topics, tasks, and steps that will be covered in the subsequent content.

Transition: With this introduction, we embark on the implementation journey, where planning transforms into action, and strategies become reality. This section will guide us through the nuts and bolts of putting our project plan into practice, offering insights, challenges, and solutions that lie ahead.

8.2 Tools and Technologies Used

Libraries :-
Pandas: Pandas is an open-source library that is made mainly for work-

ing with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library.

NumPy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

import cv2: All packages contain Haar cascade files. cv2.data.haarcascades can be used as a shortcut to the data folder.

Pillow: Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors.

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release."[30] No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.

Anaconda: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages[citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation[citation needed]. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

Spyder

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your own software.

Features

- Editor

  Work efficiently in a multi-language editor with a function/class browser, real-time code analysis tools (pyflakes, pylint, and pycodestyle), automatic code completion (jedi and rope), horizontal/vertical splitting, and go-to-definition.

- Interactive console

  Harness the power of as many IPython consoles as you like with full workspace and debugging support, all within the flexibility of a full GUI interface. Instantly run your code by line, cell, or file, and render plots right inline with the output or in interactive windows.

- Documentation viewer

  Render documentation in real-time with Sphinx for any class or function, whether external or user-created, from either the Editor or a Console.

  Supervised machine learning models are trained with labeled data sets, which allow the models to learn and grow more accurate over time. For example, an algorithm would be trained with pictures of dogs and other things, all labeled by humans, and the machine would learn ways to identify pictures. There are many applications and companies that used machine learning for doing their day to day process as it is being more accurate and precise than manual interventions. Simply put, machine learning allows the user to feed a computer algorithm an immense amount of data and have the computer analyze and make data-driven recommendations and decisions based on only the input data.

8.3 Algorithm

Algorithm : Logistic regression

- Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.Logistic regression is a statistical and machine learning model used for binary classification and, with some modifications, for multiclass classification problems. It's a type of regression analysis that is well-suited for predicting the probability of an event occurring (e.g., a customer making a purchase, a patient having a particular disease) based on one or more predictor variables.

Chapter 9

Software Testing

9.1 Purpose

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software testing can also be stated as the process of validating and verifying that a software program or application or product:
1. Meets the business and technical requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics

9.2 Scope of testing

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted. Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

9.3 Software to be tested

1. Edraw Max:

It enables students, teachers and business professional store liable create and publish various kinds of diagram store present any ideas. With this application users can easily create professional- looking flow charts, organizational charts, network diagrams, business presentations, building plans, mind maps, science illustration, fashion designs, UML diagrams and much more.

2. Star UML:

Star UML is a fully fledged, open source, UML modeling tool thats supports the ability to create software designs, from basic concepts, through to the coded solution. The user should be aware that this tool is more complex than a simple UML diagram editing tool, in that, through the use of the model Drive Architecture (MDA) standard, the tool supports complex modeling which is realizable in code.

9.4 Test Plan

To test this application we are going with proper sequencing of testing like unit, integration, validation, GUI, Low level and High level test cases, major scenarios likewise. We will go with the GUI testing first and then integration testing. After integration testing performs the high level test cases and major scenarios which can affect the working on the application. We will perform the testing on the data transmitted using the various inputs and outputs and validate the results. It also intends to cover any deviations that the project might take from the initially agreed Test Strategy in terms of scope, testing methodology, tools, etc.. This test plan covers details of testing activities for this project and scope.

9.5 Type of testing

9.5.1 Unit Testing

It is the testing of individual software units of the application .it is done after the complexion of an individual unit before integration. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process,application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.5.2 Regression Testing

Regression testing is a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements. Regression testing is responsible for the overall stability and functionality of the existing features.

9.5.3 Smoke Testing

Smoke Testing comes into the picture at the time of receiving build software from the development team. The purpose of smoke testing is to determine whether the build software is testable or not. It is done at the time of ”building software.” This process is also known as ”Day 0”.

It is a time-saving process. It reduces testing time because testing is done only when the key features of the application are not working or if the key bugs are not fixed. The focus of Smoke Testing is on the workflow of the core and primary functions of the application.

9.5.4 System Testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.
Integration Testing
Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Testing Strategy
Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

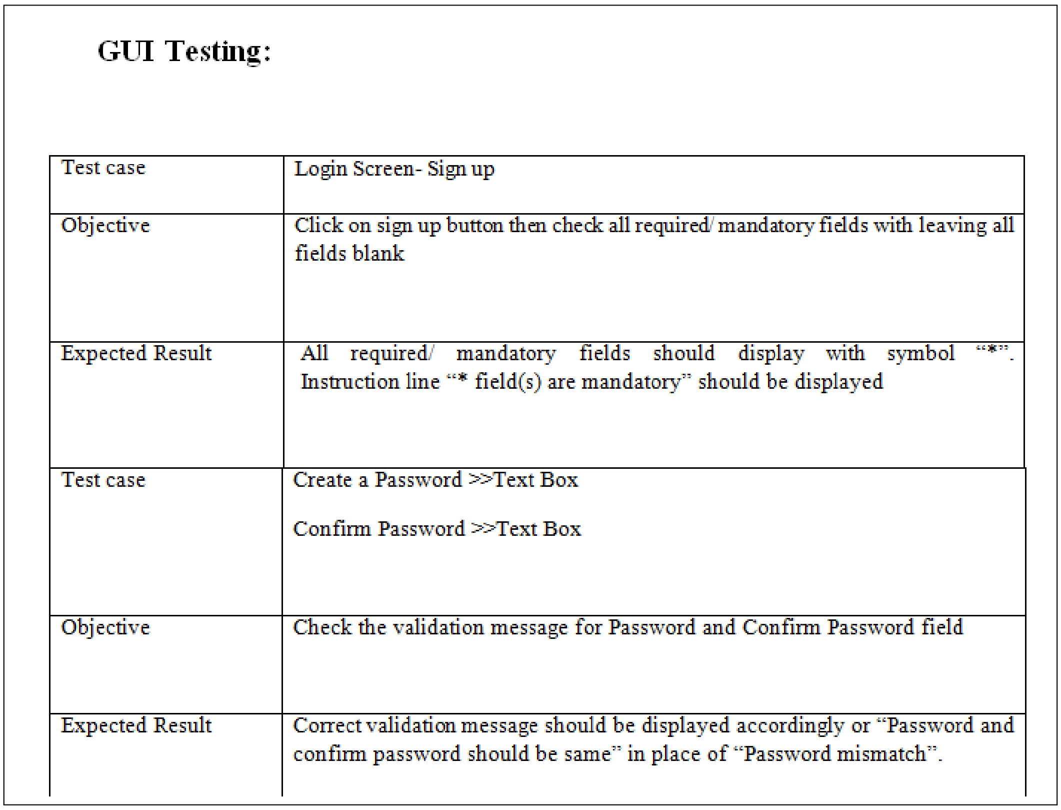
1. White-box testing

   In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.
2. Black-box testing

   Black-box testing treats the software as a quot;black boxquot;, examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it.
3. Grey-box testing

   Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software.

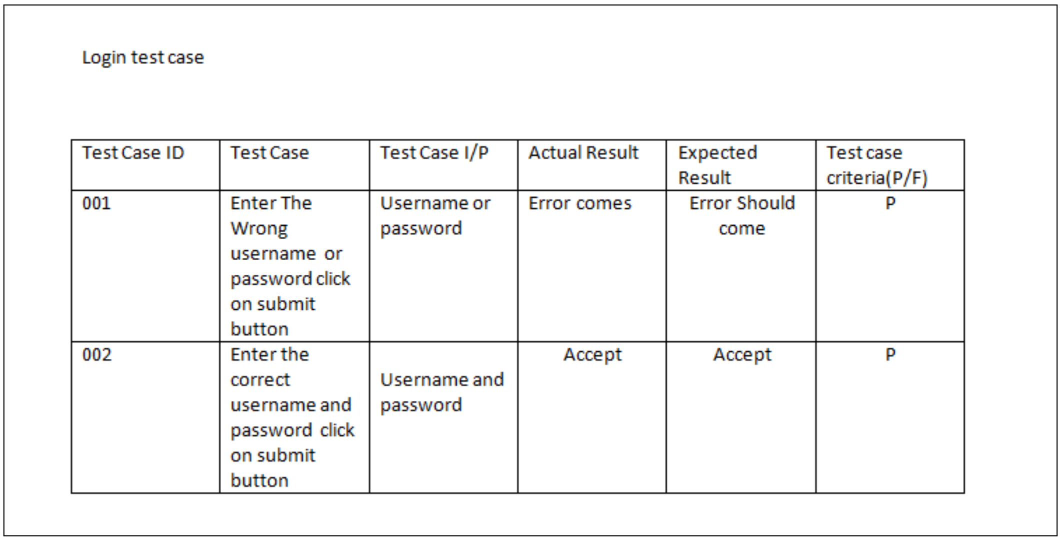
9.6 Test Cases



GUI Testing:

Test case	Login Screen- Sign up
Objective	Click on sign up button then check all required/ mandatory fields with leaving all fields blank
Expected Result	All required/ mandatory fields should display with symbol "*". Instruction line "* field(s) are mandatory" should be displayed
Test case	Create a Password >>Text Box Confirm Password >>Text Box
Objective	Check the validation message for Password and Confirm Password field
Expected Result	Correct validation message should be displayed accordingly or "Password and confirm password should be same" in place of "Password mismatch".

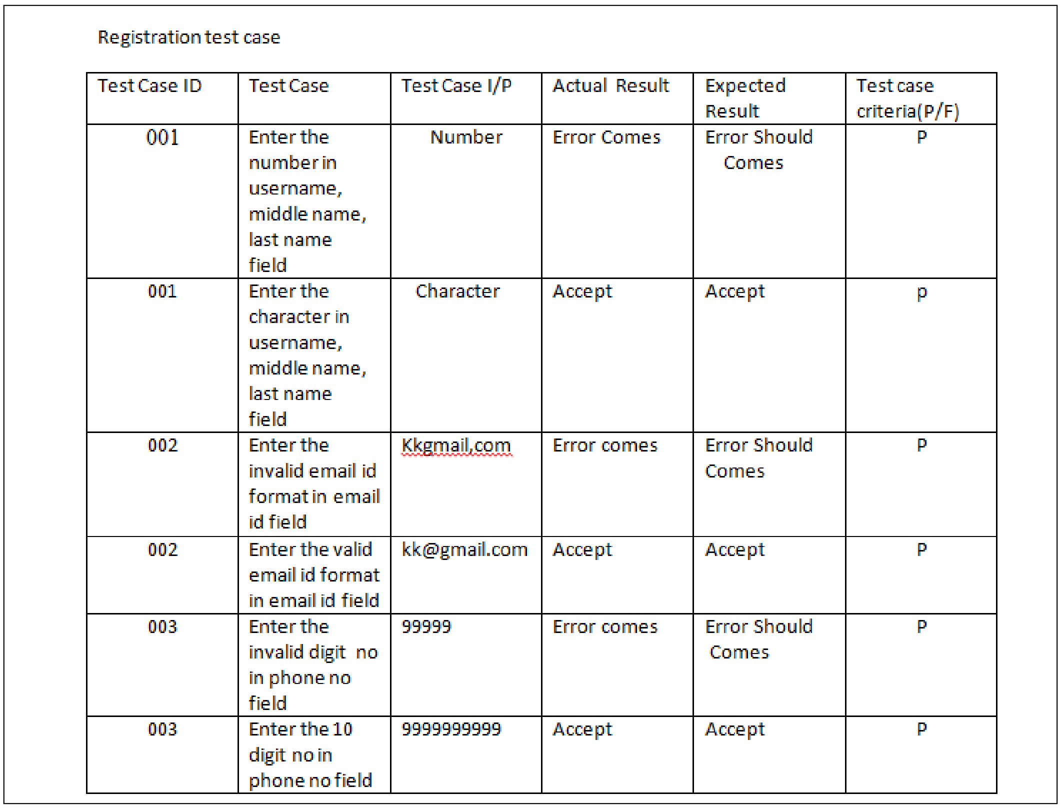
Figure 9.1: GUI Testing



Login test case

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Enter The Wrong username or password click on submit button	Username or password	Error comes	Error Should come	P
002	Enter the correct username and password click on submit button	Username and password	Accept	Accept	P

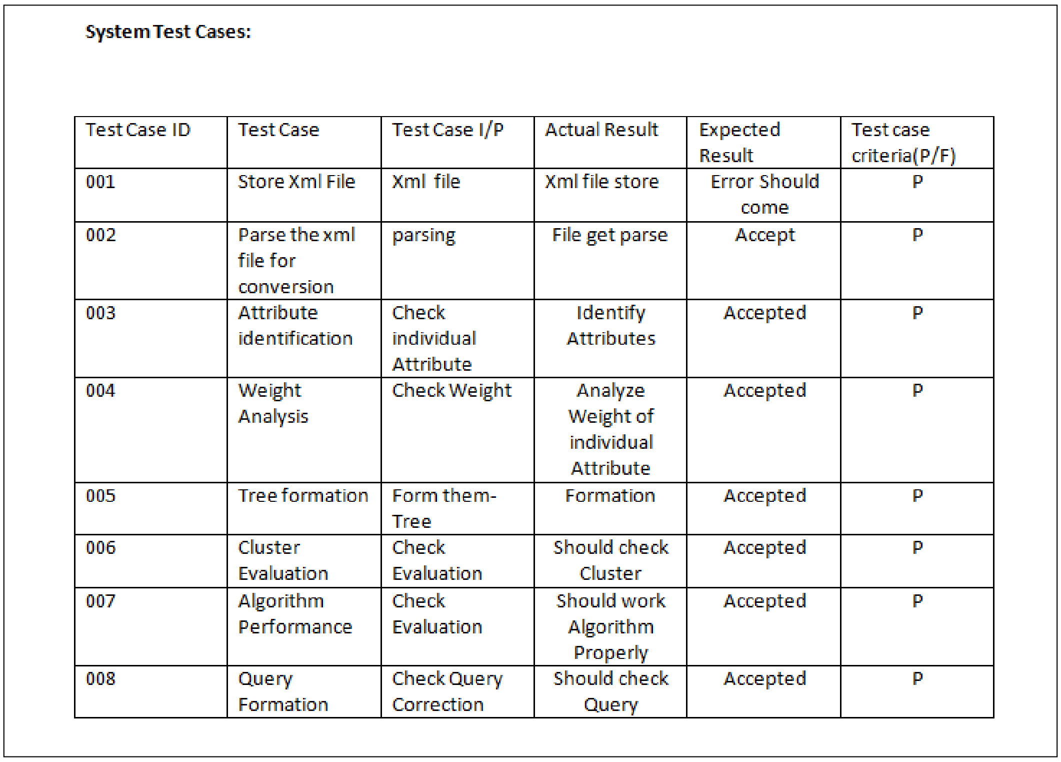
Figure 9.2: login test Cases



Registration test case

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Enter the number in username, middle name, last name field	Number	Error Comes	Error Should Comes	P
001	Enter the character in username, middle name, last name field	Character	Accept	Accept	p
002	Enter the invalid email id format in email id field	Kkgmail,com	Error comes	Error Should Comes	P
002	Enter the valid email id format in email id field	kk@gmail.com	Accept	Accept	P
003	Enter the invalid digit no in phone no field	99999	Error comes	Error Should Comes	P
003	Enter the 10 digit no in phone no field	9999999999	Accept	Accept	P

Figure 9.3: Registration Test cases



System Test Cases:

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Store Xml File	Xml file	Xml file store	Error Should come	P
002	Parse the xml file for conversion	parsing	File get parse	Accept	P
003	Attribute identification	Check individual Attribute	Identify Attributes	Accepted	P
004	Weight Analysis	Check Weight	Analyze Weight of individual Attribute	Accepted	P
005	Tree formation	Form them-Tree	Formation	Accepted	P
006	Cluster Evaluation	Check Evaluation	Should check Cluster	Accepted	P
007	Algorithm Performance	Check Evaluation	Should work Algorithm Properly	Accepted	P
008	Query Formation	Check Query Correction	Should check Query	Accepted	P

Figure 9.4: System Test Cases

Chapter 10

Deployment and Maintenance

10.1 Deployment and Maintenance

Deployment and maintenance are critical phases in the lifecycle of a software application, system, or product. In this section, you can outline the processes and considerations related to deploying your product and ensuring its ongoing maintenance and support. Here's a general structure and content for a "Deployment and Maintenance" section

10.2 User help

Begin with an introduction that explains the purpose of this section, which is to offer support and guidance to users. Mention that user help is essential for a positive user experience.Provide a subsection that helps users get started with your product or service. This could include information on how to create an account, install software, or begin using the product.Mention if there's an online community or user forum where users can seek help from other users. Community support can be valuable for sharing experiences and tips.

Chapter 11

Conclusion and future scope

- The process described in this research paper proposes a technique for automatically analyzing cricket shots that uses deep learning strategies to reach a significantly better result. LR are used in the described technique to utilize the video input of each cricket shot being played. A dataset of numerous distinct shots performed by a batsman is used to train the LR model. The dataset is first preprocessed, and the preprocessed pictures are then normalized before being fed into the LR model for training. Once the trained model is complete, it is utilized to do the testing utilizing input video that has been adequately preprocessed and normalized without first being exposed to LR detection. To perform the cricket shot assessment, the results are efficiently categorized utilizing the decision-making process.

Chapter 12

References

Annexure A

References

1 [[1] A. Javed, K. B. Bajwa, H. Malik, and A. Irtaza, ”An Efficient Framework for Automatic Highlights Generation from Sports Videos,” in IEEE Signal Processing Letters, vol. 23, no. 7, pp. 954-958, July 2016, DOI: 10.1109/LSP.2016.2573042.

2 [2] A S. Rao, J. Gubbi, S. Marusic, and M. Palaniswami, ”Crowd Event Detection on Optical Flow Manifolds,” in IEEE Transactions on Cybernetics, vol. 46, no. 7, pp. 1524-1537, July 2016, DOI: 10.1109/TCYB.2015.2451136.

3 [3] D. Tang, ”Hybridized Hierarchical Deep Convolutional Neural Network for Sports Rehabilitation Exercises,” in IEEE Access, vol. 8, pp. 118969-118977, 2020, DOI: 10.1109/ACCESS.2020.3005189.

4 [4] Følstad, Asbjørn, and Marita Skjuve. ”Chatbots for customer service: user experience and motivation.” In Proceedings of the 1st International Conference on Conversational User Interfaces, pp. 1-9. 2019

5 [5] H. Ma and X. Pang, ”Research and Analysis of Sports Medical Data Processing Algorithms Based on Deep Learning and Internet of Things,” in IEEE Access, vol. 7, pp. 118839-118849, 2019, DOI: 10.1109/ACCESS.2019.2936945.

6 [6] R. Ji, ”Research on Basketball Shooting Action Based on Image Feature Extraction and Machine Learning,” in IEEE Access, vol. 8, pp. 138743-138751, 2020, DOI: 10.1109/ACCESS.2020.3012456. e-ISSN: 2582-5208 International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open

Access, Fully Refereed International Journal) Volume:04/Issue:05/May-2022 Impact Factor- 6.752 www.irjmets.com www.irjmets.com @International Research Journal of Modernization in Engineering, Technology and Science [1830]

7 [7] M. Moness, S. K. Loutfy and M. A. Massoud, "Selecting Promising Junior Swimmers in Egypt Using Automated Biometric Algorithms of Image Processing and Fuzzy Concepts," in IEEE Access, vol. 9, pp. 89476-89496, 2021, DOI: 10.1109/ACCESS.2021.3088409.

8 [8] J. Ševčík, V. Šmídl, and F. Šroubek, "An Adaptive Correlated Image Prior for Image Restoration Problems," in IEEE Signal Processing Letters, vol. 25, no. 7, pp. 1024-1028, July 2018, DOI: 10.1109/LSP.2018.2836964.

9 [9] T. Matsui and M. Ikehara, "Single-Image Fence Removal Using Deep Convolutional Neural Network," in IEEE Access, vol. 8, pp. 38846-38854, 2020, DOI: 10.1109/ACCESS.2019.2960087.

10 [10] R. Yuan, Z. Zhang, P. Song, J. Zhang, and L. Qin, "Construction of Virtual Video Scene and Its Visualization During Sports Training," in IEEE Access, vol. 8, pp. 124999-125012, 2020, DOI: 10.1109/ACCESS.2020.3007897.

Annexure B

Laboratory assignments on Project Analysis of Algorithmic Design

- To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.
  Refer [?] for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

I	D	E	A
Increase	Drive	Educate	Accelerate
Improve	Deliver	Evaluate	Associate
Ignore	Decrease	Eliminate	Avoid

Table B.1: IDEA Matrix

- Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.
- input x,output y, y=f(x)

Annexure C

Laboratory assignments on Project Quality and Reliability Testing of Project Design

It should include assignments such as

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading
- Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.
- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.
- Additional assignments by the guide. If project type as Entreprenaur, Refer [?],[?],[?], [?]

Annexure D

Project Planner

NO	TASK	DURATION(Days)	START DATE	END DATE
1	Group Formation	4	-	-
2	Decide Area Of Interest	4	-	-
3	Search topic	5	-	-
4	Topic Selection	5	-	-
5	Section Topic	5	-	-
6	Search Related Information	12	-	-
7	Understanding concept	7	-	-
8	Search essential document(IEEE and White paper,software)	6	-	-
9	Problem Defination	2	-	-
10	Literature Survey	5	-	-
11	SRS	14	-	-
12	Project planning	2	-	-
13	Modeling and design	10	-	-
14	Technical Specification	2	-	-
15	PPT	6	-	-

Annexure E

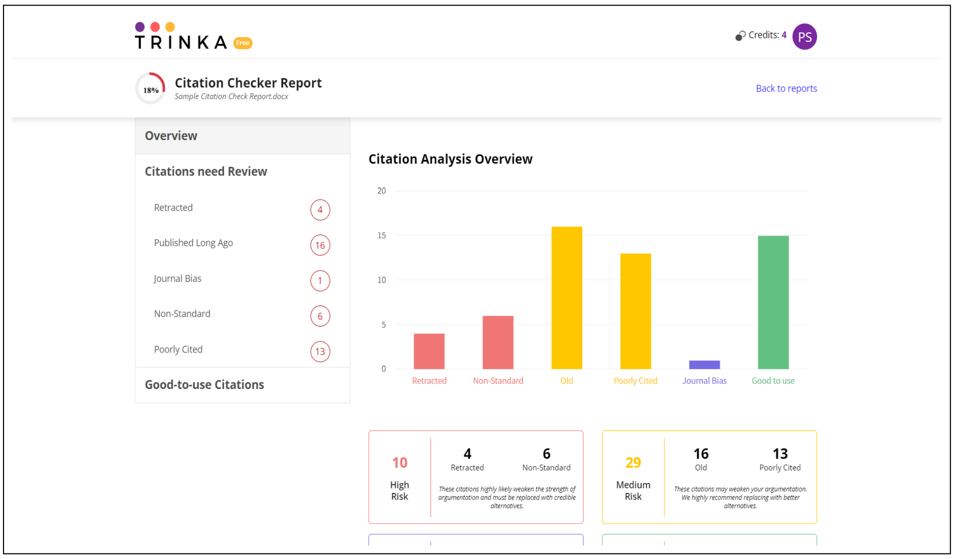
Reviewers Comments of Paper Submitted

(At-least one technical paper must be submitted in Term-I on the project design in the conferences/workshops in IITs, Central Universities or UoP Conferences or equivalent International Conferences Sponsored by IEEE/ACM)

1. Paper Title:
2. Name of the Conference/Journal where paper submitted :
3. Paper accepted/rejected :
4. Review comments by reviewer :
5. Corrective actions if any :

Annexure F

Plagiarism Report



Citation Checker Report

Sample Citation Check Report.docx

Overview

Citations need Review

Retracted	4
Published Long Ago	16
Journal Bias	1
Non-Standard	6
Poorly Cited	13

Good-to-use Citations

Citation Analysis Overview

10 High Risk — 4 Retracted, 6 Non-Standard

These citations highly likely weaken the strength of argumentation and must be replaced with credible alternatives.

29 Medium Risk — 16 Old, 13 Poorly Cited

These citations may weaken your argumentation. We highly recommend replacing with better alternatives.

Figure F.1: Plagarism

Annexure G

Term-II Project Laboratory Assignments

1. Review of design and necessary corrective actions taking into consideration the feedback report of Term I assessment, and other competitions/conferences participated like IIT, Central Universities, University Conferences or equivalent centers of excellence etc.
2. Project workstation selection, installations along with setup and installation report preparations.
3. Programming of the project functions, interfaces and GUI (if any) as per 1 st Term term-work submission using corrective actions recommended in Term-I assessment of Term-work.
4. Test tool selection and testing of various test cases for the project performed and generate various testing result charts, graphs etc. including reliability testing.
   Additional assignments for the Entrepreneurship Project:
5. Installations and Reliability Testing Reports at the client end.

1. Name : Avishkar Uttam Jadhav
2. Date of Birth : 25/07/2002
3. Gender : Male
4. Permanent Address : At Post Mukhed, Tal:Niphad, Nashik
5. E-Mail : avishkarjadhav623@gmail.com
6. Mobile/Contact No. : 8999514224
7. Placement Details : Unplaced
8. Paper Published : No

1. Name : Piyush Sandeep Sandhan
2. Date of Birth : 20/10/2002
3. Gender : Male
4. Permanent Address : At Post Nashik
5. E-Mail : sandhanpiyush20@gmail.com
6. Mobile/Contact No. : 8642821111
7. Placement Details : Unplaced
8. Paper Published : No

1. Name : Swayam Dinesh Nikam
2. Date of Birth : 2/12/2002
3. Gender : Male
4. Permanent Address : Nashik
5. E-Mail : swayamnikam1@gmail.com
6. Mobile/Contact No. : 8847722922
7. Placement Details : placed
8. Paper Published : No

1. Name : Vedant Ramnath Sadgir
2. Date of Birth : 12/12/2001
3. Gender : Male
4. Permanent Address : Nashik
5. E-Mail : vedant.sadgir12@gmail.com
6. Mobile/Contact No. : 7499034670
7. Placement Details : placed
8. Paper Published : No